# LFSR based Generation of Multicycle Test

**Fareedha**[1] **Dr.K.Seetharam**[2]
syedfareedha@gmail.com[1] seetharamkhetavath@gmail.com[2]
[1] PG Scholar, Chaitanya Institute Of Technology And Science, Kishanpura, Hanamkonda, Warangal.
[2] Associate Professor, Chaitanya Institute Of Technology And Science, Kishanpura, Hanamkonda, Warangal.

**Abstract—** The multicycle test set whose scan-in states are compressed into seeds for an LFSR, and whose primary input vectors are held constant during the application ofa multicycle test. The goal of computing multicycle tests is to provide test compaction that reduces both the test application time and the test data volume. To avoid sequential test generation, the procedure uses a single cycle test set to guide the computation of multicycle tests. The procedure optimizes every multicycle test, and increases the number of faults it detects, by adjusting its seed, primary input vector, and number of functional clock cycles. Optimizing the seed instead of the scan-instate avoids the computation of scan-in states for which seeds do not exist. Experimental results for benchmark circuits are presented to demonstrate the effectiveness of the procedure.

## I. INTRODUCTION

A direct criticism move enlist resembles a move enlist with input. The yields of a portion of the flipflops in the move of XOR door is the contribution to the primary flipflop in the principal move enlist. The underlying quality put away in the move enlist is known as the seed esteem and it can never be every one of the zeros. Contingent upon the yields input to the XOR door a LFSR produces an irregular succession of bits. Due to this property LFSRs are utilized as a part of correspondence and blunder remedy circuits for producing pseudoclamor and pseudo-irregular number arrangements and they are additionally utilized as a part of information encryption and information pressure circuits in cryptography. In this LFSR the sweep in and examine out activities of a solitary test cycle has a solitary capacity unit, while a multi cycle test has at least one number of useful clock cycles. Multi cycle test where considered as the successful test compaction and the outcome from the perception will be seen by the LFSR. Amid the practical clock cycle the combinational rationale of the circuit gets an information design that can be utilized for identifying issues. An expansive number of utilitarian clock cycles enables more blames to be recognized. Subsequently, multi cycle test may identifies more blames contrasted with single-cycle test. With more distinguished flaws for each test cycle the deficiencies will decreased. This lessens a few sweep tasks that a test set requires. With few output tasks, the information volume and the application time will be lessened. The way that each test comprises of more practical clock cycles negligibly affects the test application when the quantity of clock cycles are fortified. The information volume is autonomous of the quantity of capacity clock cycles, the essential information vector is steady amid the test. This is the basic prerequisite to address analyzer impediments that can keep the essential information vector from being changed amid a test.

## II. EXISTING WORK

The scan-in and scan-out operations of a test, a single cycle test has a single functional clock cycle, while a multicycle test has one or more functional clock cycles. Multicycle tests were considered. Their effectiveness for test compaction was demonstrated and results from the following observations. During a functional clock cycle of a test, the combinational logic of the circuit receives an input pattern that can be used for detecting faults. A larger number of functional clock cycles allows more faults to be detected. As a result, a multicycle test may detect more faults than a single-cycle test. With more detected faults for every test, the number of tests is reduced.

## III. Multicycle Test Scheme

Fig. 1 shows the proposed multi-cycle test scheme with partial observation of flip-flops. Unlike the case of [10], we use scan-based logic BIST instead of full-scan test. This is because logic BIST with re-seeding will be more feasible for field test with small memory resource than compression-based deterministic full-scan test in the current technologies. In this scheme, we target to reduce the number of seeds that is needed for achieving the given fault coverage or to improve the fault coverage for the given number of seeds. In the figure, input vectors to the combinational circuit under test (CUT) is provided to flip-flops (FFs) through scan chains from a test pattern generator (TPG), which can be a linear feedback signature register (LFSR) or a cellular Automata (CA) with re-seeding capability. The output values of CUT are captured into FFs at each clock cycle during test mode (capture mode). They are scanned out to Compactor A, which consists of an XOR based space compactor and a MISR. At the same time, a part of FFs are connected directly (i.e. without scan-out) to additional Compactor B, which also consists of an XOR based space compactor and a MISR. Note that we refer to the FFs connected to Compactor B as partial FFs in this paper. Primary inputs and outputs are isolated from the CUT using boundary scan cells in case of at-speed test. The figure is simplified to a single scan chain, but it can easily be enhanced to multiple scan chains.

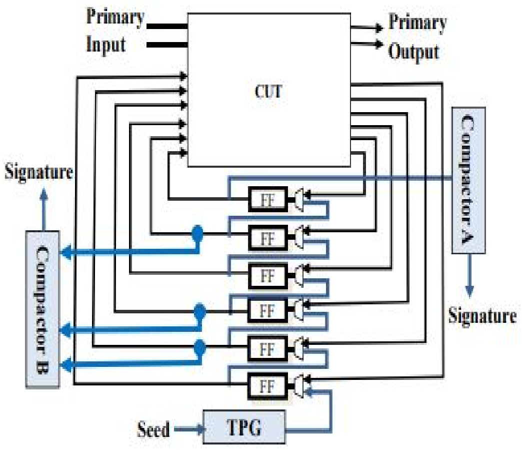


Fig.1 Block Diagram of Multicycle test Scheme

**Proposed System.**

The modification of a seed is implemented by complementing bits of si one by one, and recomputing the test ti that the LFSR produces. A bit complementation is accepted when ti satisfies certain objectives (these objectives are related to the generation of diagnostic tests). In the procedure described the bits of si and vi, as well as the value of li, are modified together in order to produce an effective multicycle test. The target faults are single stuck-at faults. The procedure is developed assuming that an LFSR is given. This is also describes a modified binary search process for selecting an LFSR out of a given set of available LFSRs.

## COMPUTING A COMPRESSED MULTI-CYCLE TESTS:

The method portrayed in this area acknowledges a solitary cycle test wi =< qi,ui,1 >, an arrangement of target deficiencies Fi, and an underlying target L for the quantity of useful check cycles in a multicycle test. It delivers a compacted multicycle test ti =< si,vi,li > that distinguishes whatever number issues from Fi as could be expected under the circumstances. To check whether wi is powerful in managing the age of a multicycle test, the system performs blame reproduction of Fi under wi. It stores the arrangement of identified blames in Di. In the event that Di = ∅, the system does not endeavor to figure a multicycle test in light of wi. It denotes that wi isn't powerful to abstain from thinking of it as again in later cycles. On the off chance that Di 6= ∅, the methodology proceeds as takes after. Not all the specified estimations of wi =< qi,ui,1 > are required for blame location. To guarantee that lone vital esteems manage the age of ti, the methodology first changes whatever number specified estimations of qi as could be expected under the circumstances into unspecified values without losing the recognition of any blame from Fi. The rest of the specified esteems are vital for the identification of target flaws. They would thus be able to be utilized for directing the age of ti. For a circuit with k state factors, let qi(j) be the estimation of state variable j, where 0 ≤ j < k. For 0 ≤ j < k, if qi(j) 6= x, the method doles out qi(j) = x, and reproduces Di under < qi, ui,1 >. On the off chance that every one of the shortcomings in Di are distinguished, the system acknowledges the unspecified estimation of qi(j). Else, it reestablishes its past specified esteem. To process ti =< si,vi,li >, the technique introduces si haphazardly, and appoints vi = ui and li = L. Give pi a chance to be the output in express that si produces. The strategy mimics Fi under < pi,vi,li >, and stores the quantity of distinguished blames in a variable that is signified by dbest. What's more, it registers the Hamming separation amongst pi and qi, and stores it in a variable that is meant by hbest. The Hamming separation is equivalent to the quantity of state factors j where qi(j) 6= x and pi(j) 6= qi(j). As ti is modified, dbest stores the biggest number of recognized shortcomings, and hbest stores the littlest Hamming separation got with the biggest number of identified deficiencies. The objective of altering ti is to build the quantity of distinguished flaws (or the estimation of dbest), and lessen the Hamming separation amongst pi and qi (or the estimation of hbest). Expanding the quantity of distinguished shortcomings is given a higher need. On the off chance that the methodology can't expand the quantity of recognized deficiencies, decreasing the Hamming separation amongst pi and qi may in the long run enable ti to identify shortcomings from Di. The modification of ti is refined in three stages that are connected iteratively. The first step endeavors to supplement bits of si. The second step endeavors to supplement bits of vi. The third step endeavors to supplant li with an alternate an incentive from the set {1,2,...,LMAX}, where LMAX is a steady upper bound on li. Amid the first step, the method thinks about all of si. With a B-bit LFSR, the method considers si(j) for 0 ≤ j < B. At the point when the methodology considers si(j), it supplements its esteem, and recomputed the output in state pi of ti. It recreates Fi under ti, and stores the quantity of distinguished blames in a variable that is meant by di. Also, it figures the Hamming separation amongst pi and qi, and stores it in a variable that is indicated by howdy. The methodology acknowledges the complementation of si(j) if di > dbest, or di = dbest and hello there < hbest. In this way, to acknowledge the complementation of si(j), the technique requires either an expansion in the quantity of recognized shortcomings, or a lessening in the Hamming separation with a similar number of distinguished flaws. In the event that this condition is satis the strategy refreshes dbest and hbest by relegating = di and hbest = howdy. Something else, the technique reestablishes the past estimation of si(j) by dbest supplementing it once more.

A comparative procedure is connected to vi, with the exception of that supplementing bits of vi does not influence the Hamming separation amongst pi and qi. The same applies to li. For li, the technique considers distinctive quantities of practical clock cycles, which are given by lnew = LMAX, LMAX , ..., 1, in a specific order. In the event that lnew 6= li, the method doles out li = lnew. It reenacts Fi under ti, and stores the quantity of identified blames in di. The methodology acknowledges the new dbest. For this situation, it doles out dbest = di. Else, it reestablishes li to its past esteem. This procedure lean towards a lower estimation of li in the event that it doesn't diminish the quantity of recognized issues. The quantity of cycles of the three stages is a steady that is meant by NMOD. After NMOD cycles the technique restores the test ti, and the quantity of flaws that it recognizes, dbest. The strategy for processing ti is abridged straightaway. For consistency, the Hamming separation amongst pi and qi is d for si, vi and li despite the fact that it can't be influenced by altering vi or li. The quantity of essential data sources is meant by n.

Methodology 1: Computing a packed multicycle test ti 1) Simulate Fi under < qi,ui,1 > and arrangement of distinguished shortcomings, Di. On the off chance that Di = ∅, dole out use(wi) = 0, and return dbest = 0.

2) Unspecify qi with the end goal that wi would keep on detecting every one of the flaws in Di.

3) Specify si haphazardly. Allocate vi = ui and li = L.

4) Compute pi. Recreate Fi under ti and allocate the quantity of distinguished shortcomings to dbest. Process the Hamming separation amongst pi and qi, and dole out it to hbest.

5) For nmod = 0, 1, ..., NMOD : a) For j = 0, 1, ..., B − 1:

I) Complement si(j). Call Procedure accept_ mod (). On the off chance that the strategy returns FALSE, supplement si(j) once more. b) For j = 0, 1, ..., n − 1:

I) Complement vi(j). Call Procedure accept_ mod (). On the off chance that the strategy returns FALSE, supplement vi(j) once more .

c) For lnew = LMAX, LMAX − 1, ..., 1, if li 6= lnew i) Assign li = lnew. Call Procedure accept_ mod (). On the off chance that the strategy returns FALSE, reestablish the past estimation of li. 6) Return ti and dbest. Methodology accept_ mod() 1) Compute pi. Reproduce Fi under ti and allot the quantity of identified issues to di.

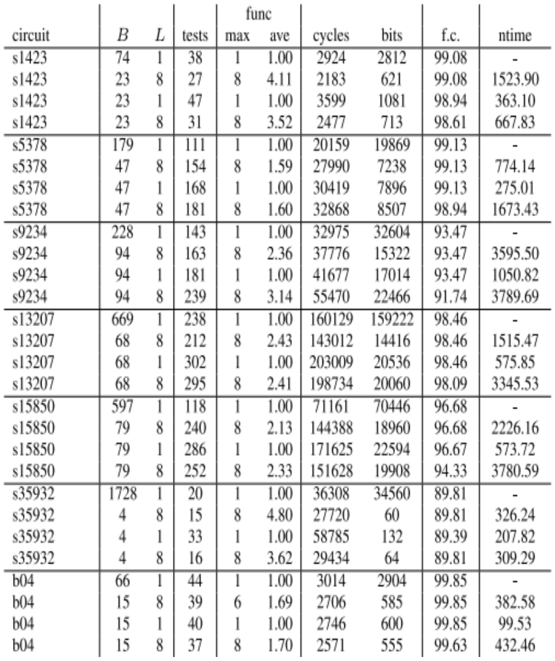
2) Compute the Hamming separation amongst pi and qi, and allot it to hello there.

3) If di > dbest, or di = dbest and hello there ≤ hbest, dole out dbest = di and hbest = hey, and return TRUE. 4) Return FALSE. Strategy 1 performs NMOD emphasess where it considers B bits of si, n bits of vi, and LMAX − 1 choices for li. For each situation it reproduces one modified test, for an aggregate of NMOD(B+n+LMAX−1) tests.

## IV RESULTS

The simulation and verified effect are shown, once the realistic verification is done, the RTL mannequin is taken to the synthesis method using the

Xilinx ISE instrument. This design is synthesized and its outcome were analyzed as follows.



| | | | | func | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| circuit | B | L | tests | max | ave | cycles | bits | f.c. | ntime |
| s1423 | 74 | 1 | 38 | 1 | 1.00 | 2924 | 2812 | 99.08 | - |
| s1423 | 23 | 8 | 27 | 8 | 4.11 | 2183 | 621 | 99.08 | 1523.90 |
| s1423 | 23 | 1 | 47 | 1 | 1.00 | 3599 | 1081 | 98.94 | 363.10 |
| s1423 | 23 | 8 | 31 | 8 | 3.52 | 2477 | 713 | 98.61 | 667.83 |
| s5378 | 179 | 1 | 111 | 1 | 1.00 | 20159 | 19869 | 99.13 | - |
| s5378 | 47 | 8 | 154 | 8 | 1.59 | 27990 | 7238 | 99.13 | 774.14 |
| s5378 | 47 | 1 | 168 | 1 | 1.00 | 30419 | 7896 | 99.13 | 275.01 |
| s5378 | 47 | 8 | 181 | 8 | 1.60 | 32868 | 8507 | 98.94 | 1673.43 |
| s9234 | 228 | 1 | 143 | 1 | 1.00 | 32975 | 32604 | 93.47 | - |
| s9234 | 94 | 8 | 163 | 8 | 2.36 | 37776 | 15322 | 93.47 | 3595.50 |
| s9234 | 94 | 1 | 181 | 1 | 1.00 | 41677 | 17014 | 93.47 | 1050.82 |
| s9234 | 94 | 8 | 239 | 8 | 3.14 | 55470 | 22466 | 91.74 | 3789.69 |
| s13207 | 669 | 1 | 238 | 1 | 1.00 | 160129 | 159222 | 98.46 | - |
| s13207 | 68 | 8 | 212 | 8 | 2.43 | 143012 | 14416 | 98.46 | 1515.47 |
| s13207 | 68 | 1 | 302 | 1 | 1.00 | 203009 | 20536 | 98.46 | 575.85 |
| s13207 | 68 | 8 | 295 | 8 | 2.41 | 198734 | 20060 | 98.09 | 3345.53 |
| s15850 | 597 | 1 | 118 | 1 | 1.00 | 71161 | 70446 | 96.68 | - |
| s15850 | 79 | 8 | 240 | 8 | 2.13 | 144388 | 18960 | 96.68 | 2226.16 |
| s15850 | 79 | 1 | 286 | 1 | 1.00 | 171625 | 22594 | 96.67 | 573.72 |
| s15850 | 79 | 8 | 252 | 8 | 2.33 | 151628 | 19908 | 94.33 | 3780.59 |
| s35932 | 1728 | 1 | 20 | 1 | 1.00 | 36308 | 34560 | 89.81 | - |
| s35932 | 4 | 8 | 15 | 8 | 4.80 | 27720 | 60 | 89.81 | 326.24 |
| s35932 | 4 | 1 | 33 | 1 | 1.00 | 58785 | 132 | 89.39 | 207.82 |
| s35932 | 4 | 8 | 16 | 8 | 3.62 | 29434 | 64 | 89.81 | 309.29 |
| b04 | 66 | 1 | 44 | 1 | 1.00 | 3014 | 2904 | 99.85 | - |
| b04 | 15 | 8 | 39 | 6 | 1.69 | 2706 | 585 | 99.85 | 382.58 |
| b04 | 15 | 1 | 40 | 1 | 1.00 | 2746 | 600 | 99.85 | 99.53 |
| b04 | 15 | 8 | 37 | 8 | 1.70 | 2571 | 555 | 99.63 | 432.46 |

## V CONCLUSION

This paper depicted a technique for figuring a multicycle test set with the accompanying properties: 1) the output in states are compacted into seeds for a LFSR and 2) the essential information vectors are held steady amid the utilization of a multicycle test. The methodology is guided by a single cycle test set. This test set does not need to be relevant utilizing a LFSR with a predetermined number of bits. The system modifies an at first arbitrary seed, the essential information vector, and the quantity of practical clock cycles of each multicycle test to distinguish the biggest conceivable number of shortcomings. This procedure is guided by a solitary cycle test. Exploratory outcomes for benchmark circuits showed the viability of multicycle tests in accomplishing test compaction when the tests are required to be producible by a LFSR with a specific end goal to accomplish test information pressure.

## REFERENCES

[1] S. Y. Lee and K. K. Saluja, "Test application time reduction for sequential circuits with scan," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 14, no. 9, pp. 1128–1140, Sep. 1995.

[2] I. Pomeranz and S. M. Reddy, "Static test compaction for scan-based designs to reduce test application time," in Proc. Asian Test Symp., Singapore, 1998, pp. 198–203.

[3] P. C. Maxwell, R. C. Aitken, K. R. Kollitz, and A. C. Brown, "IDDQ and AC scan: The war against unmodelled defects," in Proc. Int. Test Conf., Washington, DC, USA, 1996, pp. 250–258.

[4] J. Rearick, "Too much delay fault coverage is a bad thing," in Proc. Int. Test Conf., Baltimore, MD, USA, 2001, pp. 624–633.

[5] X. Lin and R. Thompson, "Test generation for designs with multiple clocks," in Proc. Design Autom. Conf., Anaheim, CA, USA, 2003, pp. 662–667.

[6] G. Bhargava, D. Meehl, and J. Sage, "Achieving serendipitous N-detect mark-offs in multi-capture-clock scan patterns," in Proc. Int. Test Conf., Santa Clara, CA, USA, 2007, pp. 1–7.

[7] I. Park and E. J. McCluskey, "Launch-on-shift-capture transition tests," in Proc. Int. Test Conf., Santa Clara, CA, USA, 2008, pp. 1–9.

[8] E. K. Moghaddam, J. Rajski, S. M. Reddy, and M. Kassab, "At-speed scan test with low switching activity," in Proc. VLSI Test Symp., Santa Cruz, CA, USA, 2010, pp. 177–182.

[9] I. Pomeranz, "Generation of multi-cycle broadside tests," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 30, no. 8, pp. 1253–1257, Aug. 2011.

[10] I. Pomeranz, "Multicycle tests with constant primary input vectors for increased fault coverage," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 31, no. 9, pp. 1428–1438, Sep. 2012.

[11] I. Pomeranz, "Multi-cycle broadside tests with runs of constant primary input vectors," IET Comput. Digit. Tech., vol. 8, no. 2, pp. 90–96, Mar. 2014.

[12] I. Pomeranz, "A multicycle test set based on a two-cycle test set with constant primary input vectors," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 34, no. 7, pp. 1124–1132, Jul. 2015.

[13] B. Koenemann, "LFSR-coded test patterns for scan designs," in Proc. Eur. Test Conf., Munich, Germany, 1991, pp. 237–242.

[14] S. Hellebrand, S. Tarnick, J. Rajski, and B. Courtois, "Generation of vector patterns through reseeding of multiple-polynomial linear feedback shift register," in Proc. Int. Test Conf., Baltimore, MD, USA, 1992, pp. 120–129.

[15] C. Barnhart et al., "OPMISR: The foundation for compressed ATPG vectors," in Proc. Int. Test Conf., Baltimore, MD, USA, 2001, pp. 748–757.

[16] J. Rajski et al., "Embedded deterministic test for low cost manufacturing test," in Proc. Int. Test Conf., Baltimore, MD, USA, 2002, pp. 301–310.

[17] A.-W. Hakmi, H.-J. Wunderlich, C. G. Zoellin, A. Glowatz, and F. Hapke, "Programmable deterministic built-in self-test," in Proc. Int. Test Conf., Santa Clara, CA, USA, 2007, pp. 1–9.

[18] D. Czysz et al., "Deterministic clustering of incompatible test cubes for higher power-aware EDT compression," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 30, no. 8, pp. 1225–1238, Aug. 2011.

[19] A. Chandra, J. Saikia, and R. Kapur, "Breaking the test application time barriers in compression: Adaptive scan-cyclical (AS-C)," in Proc. Asian Test Symp., New Delhi, India, 2011, pp. 432–437.

[20] O. Acevedo and D. Kagaris, "Using the Berlekamp–Massey algorithm to obtain LFSR characteristic polynomials for TPG," in Proc. Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst., Austin, TX, USA, 2012, pp. 233–238.

[21] X. Lin and J. Rajski, "On utilizing test cube properties to reduce test data volume further," in Proc. Asian Test Symp., Niigata, Japan, 2012, pp. 83–88.

[22] T. Moriyasu and S. Ohtake, "A method of one-pass seed generation for LFSR-based deterministic/pseudo-random testing of static faults," in Proc. Latin-Amer. Test Symp., Puerto Vallarta, Mexico, 2015, pp. 1–6.

[23] I. Pomeranz, “Computation of seeds for LFSR-based diagnostic test generation,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 34, no. 12, pp. 2004–2012, Dec. 2015.

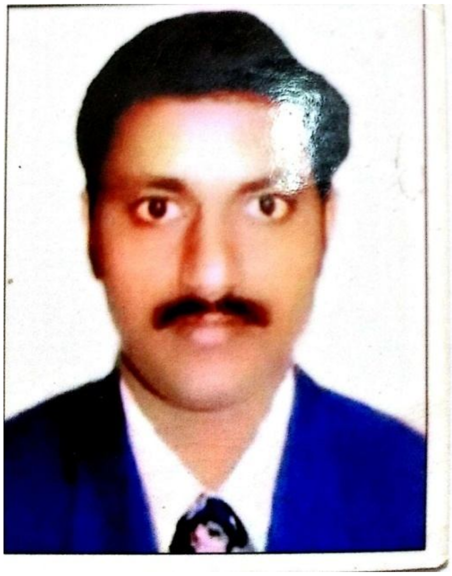
[24] I. Pomeranz, “Computing seeds for LFSR-based test generation from nontest cubes,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 24, no. 6, pp. 2392–2396, Jun. 2016.

[25] P. H. Bardell, W. H. McAnney, and J. Savir, Built-In Test for VLSI: Pseudorandom Techniques. New York, NY, USA: Wiley, 1987.

[26] I. Pomeranz and S. M. Reddy, “Forming N-detection test sets without test generation,” ACM Trans. Design Autom., vol. 12, no. 2, 2007, Art. no. 18.

**BIOGRAPHIES:**

Fareedha is perusing PG in VLSI SYSTEM DESIGN from Chaitanya Institute Of Technology And Science, Kishanpura, Hanamkonda, Warangal.



Dr. K. Seetharam is working as an Associate Professor, in Chaitanya Institute Of Technology And Science, Kishanpura, Hanamkonda, Warangal